XP-002317545

Building Hypertext Links By Computing Semantic Similarity

Stephen J. Green

Abstract—Most current automatic hypertext generation systems rely on term repetition to calculate the relatedness of two documents. There are well-recognized problems with such approaches, most notably, a vulnerability to the effects of synonymy (many words for the same concept) and polysemy (many concepts for the same word). We propose a novel method for automatic hypertext generation that is based on a technique called *lexical chaining*, a method for discovering sequences of related words in a text. This method uses a more general notion of document relatedness, and attempts to take into account the effects of synonymy and polysemy. We also present the results of an empirical study designed to test this method in the context of a question answering task from a database of newspaper articles.

Index Terms—Automatic hypertext generation, information retrieval, semantic relatedness, lexical semantics, lexical chaining.

1 Introduction

There is no question that building and maintaining a large Web site requires large amounts of time and money [1]. Aside from these concerns, there is evidence that when humans construct hypertext links they do so inconsistently; that is, different people will tend to place different links into the same document (see, for example, [2], [3]). This inconsistency may mean that the links would be less useful for a user searching for specific information. The inconsistency and high cost of manually constructing hypertext links does not necessarily mean that large-scale hypertexts (e.g., a large online newspaper) can never be built, it simply means we need to turn to automatically generated hypertext links.

Generally speaking, there are two broad categories of links that we would like to be able to automatically construct: *structural links* and *semantic links*. Structural links are those that connect the parts of a document on the basis of its logical structure, for example, the entries in a table of contents could be linked to the corresponding sections and subsections of the document (see [4] for a good example of such approaches). Semantic links, on the other hand, are links that connect documents or parts of documents on the basis of their semantic similarity. For example, the introduction to a technical report could contain links from the paragraphs describing various aspects of the work to the sections where those aspects are explained in greater detail. Semantic links can be used to connect documents (or parts of documents) when there is no explicitly specified relationship between them.

Our work focuses on this second category of links. By using a technique called *lexical chaining* [5], we can extract sets of semantically related words from text. In the sections that follow, we will describe methods for automatically generating hypertext links both within and between documents on the basis of the semantic similarity of the words that they contain. In addition, we will describe the results of an experiment that tests the proposed hypertext generation methodology against a methodology based on a traditional IR system.

• *The author is with the Microsoft Research Institute, Division of Information and Communication Sciences, Macquarie University, North Ryde, NSW 2109, Australia. E-mail: sjgreen@mri.mq.edu.au.*

Manuscript received 1 July 1998; revised 28 Oct. 1998.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 108988.

2 Related Work

Of course, we are not the first to attempt the automatic generation of hypertext links. Many of the early systems were intended for use within a single large document, rather than a large collection of documents, and focused on the construction of structural links alone. More recently, research has turned to large document collections [6], [7] and the construction of semantic links.

Automatic generation of semantic links is often treated as a special case of the more general information retrieval (IR) problem (for a recent example, see [7]). The basic premise underlying traditional IR systems is that documents that are related will use the same words. If two documents share enough terms, then we can say that they are related and should therefore have a link placed between them.

Two linguistic factors can affect this operation: *synonymy* (many words referring to the same concept, for example, *dog* and *hound*) and *polysemy* (many concepts being expressed by the same word, for example, *bank*). The impact of synonymy is that documents that use words that are synonyms of one another will not be considered related or at best will be considered to be less related than they actually are. Polysemy will have the opposite effect, causing documents that use the same word in different senses to be considered related when they should not be. Others have tried to account for these factors in IR systems and met with limited success (cf. Voorhees' work on query expansion [8]).

Our work will address these factors by using a technique called *lexical chaining* [5] drawn from the field

1041-4347/99/$10.00 © 1999 IEEE

BEST AVAILABLE COPY

RNSDOCID: <XP 2317545A 1 >

of Computational Linguistics. In addition, we propose the use of a more general notion of relatedness, one that is based on semantic similarity rather than simple term repetition.

3 Lexical Chains

A *lexical chain* is a sequence of semantically related words in a text. For example, if a text contained the words *apple* and *fruit*, they would appear in a chain together since *apple* is a kind of *fruit*. Generally speaking, a document will contain many such chains, each of which captures a portion of the cohesive structure of the document. *Cohesion* is what, as Halliday and Hasan [9] defined it, helps a text "hang together as a whole." The lexical chains contained in a text will tend to delineate the parts of the text that are "about" the same thing. Morris and Hirst [5] showed that the organization of the lexical chains in a document mirrors, in some sense, the discourse structure of that document.

The lexical chains in a text can be identified using any lexical resource that relates words by their meaning. While the original work was done using *Roget's Thesaurus* [10], our current lexical chainer, which is similar to the one described in [11], uses the WordNet database [12]. The WordNet database is composed of synonym sets or *synsets*. Each synset contains one or more words that have the same (or nearly the same) meaning. A word may appear in many synsets, depending on the number of senses that it has. Synsets can be connected to each other by several different types of links that indicate different relations. For example, two synsets can be connected by a HYPERNYM link, which indicates that the words in the source synset are instances of the words in the target synset.

For the purposes of lexical chaining, each type of link between WordNet synsets is assigned a direction of up, down, or horizontal. Upward links correspond to generalization: For example, an upward link from *apple* to *fruit* indicates that *fruit* is more general than *apple*. Downward links correspond to specialization: For example, a link from *fruit* to *apple* would have a downward direction. Horizontal links narrowly specify the senses of the synsets that they connect. The ANTONYMY relation in WordNet is considered to have a horizontal direction since it specifies the sense of a word very accurately. For example, if a text contains the words *board* and *disembark*, then it is very likely that they are being used in the senses in which they are antonyms.

Given these types of links, three kinds of relations are built between words:

1. *Extra strong*. An extra strong relation is said to exist between repetitions of the same word.
2. *Strong*. A strong relation is said to exist between words that are in the same WordNet synset (i.e., words that are synonyms). Fig. 1 shows such a relation between *person* and *human*. Strong relations are also said to exist between words that have synsets connected by a single horizontal link, as do *successor* and *predecessor* in Fig. 2, or words that have synsets connected by a single IS-A or INCLUDES relation, as do *school* and *private school* in Fig. 3.

Fig. 1. Strong relation due to synonymy.

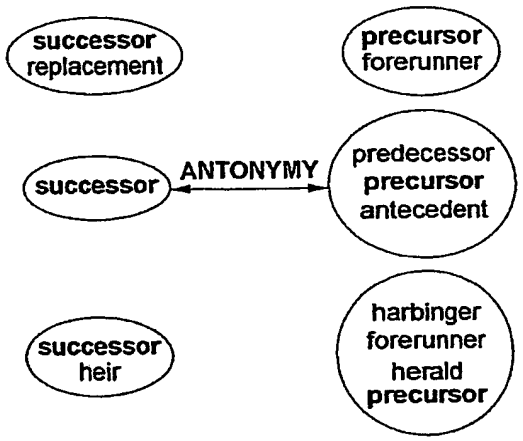

Fig. 2. Strong relation due to horizontal link.

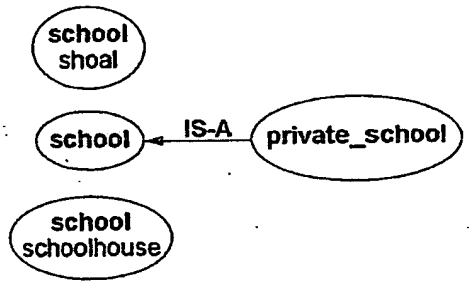

Fig. 3. Strong relation due to single link.

3. *Regular*. A regular relation is said to exist between two words when there is at least one *allowable* path between a synset containing the first word and a synset containing the second word in the WordNet database. A path is allowable if it is shorter than a given length (usually four) and adheres to three rules:

BNSDOCID: <XP_____2317545A__I_>

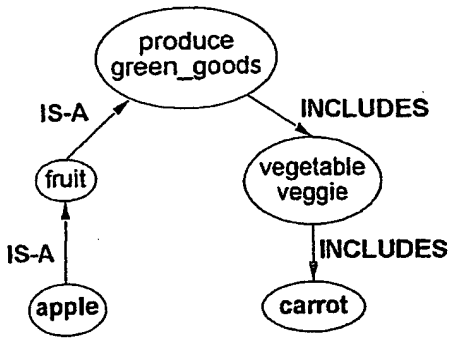

Fig. 4. A regular relation connecting *apple* and *carrot*.

- No other direction may precede an upward link;
- No more than one change of direction is allowed; and
- A horizontal link may be used to move from an upward to a downward direction.

Fig. 4 shows the regular relation that can be built between *apple* and *carrot*.

These relations specify the possible relations between words more explicitly than those used in other systems based on semantic similarity [13]. Using these relations, we can recover the lexical chains in a document. Typically, there will be several chains in a document that are written to a file during the processing of a document. In addition, a file is written containing a description of which chains appear in which paragraphs of the document.

4 Linking within the Document

As mentioned earlier, Morris and Hirst [5] demonstrated that the structure of the lexical chains in a document corresponds to the structure of the document. In other words, the lexical chains will tend to delineate the parts of a document that are "about" the same topic. Due to the difficulty of building lexical chains by hand, they did not test whether this is the case for a large number of texts. If the lexical chains do indicate the structure of the document, then they are a natural tool to use when building *intradocument* links, that is, hypertext links within an document.

We decided to use paragraphs as the nodes in our hypertext. This is a natural choice, as paragraph boundaries can usually be detected easily, even in the absence of document mark-up. Fig. 5 shows paragraphs 1, 2, 5, and 8 of a news article about the trend toward "virtual parenting" [14]. Superscript numbers after a term in the text indicate to which chain a term belongs. Table 1 shows some of the lexical chains contained in this article. In this table, the column labeled *C* gives the chain number, the column labeled *Word* shows the words in the chain, and the column labeled *Syn* shows the synsets associated with a particular word after chaining has completed. The number in parentheses after a word indicates the number of times that the word appears in the document.

We will use this particular text to illustrate the process of building intradocument links. Before we begin, however, we should look at the structure of the article, in terms of how it talks about the phenomena of virtual parenting. We can do this on a paragraph-by-paragraph basis, as shown in Table 2. It is easy to see that there are some paragraphs that are related. For example, paragraph 2 (the definition of the term) is clearly related to paragraphs 5, 6, 7, 8, and 9 (examples of and warnings about virtual parenting). These are the kinds of links that we would like our method to produce.

In their original work on lexical chaining, Morris and Hirst showed a mapping between the lexical chains contained in a document and the discourse intentions (i.e., the topics the writer intends to discuss) in the document. Unfortunately, they gave no easily implementable algorithm for determining this correspondence. Furthermore, they provided no way to determine the relatedness of two parts of the document. Our goal is to provide a method to make this determination. Because this has not been attempted before, we shall try to use techniques that are as simple as possible to begin with and only turn to more complex techniques if necessary.

1	**Working**[1] **parents**[1] **note**[3]: From the **folks**[4] who brought you virtual **reality**[6] and the virtual **office**[3], now comes a new **kind**[8] of altered **state**[6] - virtual parenting.
2	Although no one is **pushing**[12] virtual-reality **headgear**[16] as a **substitute**[1] for **parents**[1], many technical ad **campaigns**[13] are promoting cellular **phones**[22], **faxes**[22], **computers**[1] and pagers to **working**[1] **parents**[1] as a way of bridging **separations**[17] from their **kids**[1]. A recent **promotion**[13] by A T & T and **Residence**[2] **Inns**[7] in the **United States**[13], for **example**[3], suggests that **business**[3] **travellers**[1] with **young**[1] children use **video**[3] and **audiotapes**[22], **voice**[3] **mail**[3], videophones and E-mail to **stay**[3] connected, including **kissing**[23] the **kids**[1] **good night**[21] by **phone**[22].
5	When **Mark**[1] Vanderbilt, a **network**[22] **systems**[22] **engineer**[1], was **planning**[19] a scientific **expedition**[13] to **Antarctica**[1], he taught his **wife**[1] and three children to send and receive live **video**[3] **feeds**[1] over the Internet.
8	More **advice**[3] from **advertisers**[1]: **Business**[3] **travellers**[1] can dine with their **kids**[1] by **speaker**[1]-phone or "tuck them in" by cordless **phone**[22]. Separately, a **management**[10] **newsletter**[24] recommends faxing your **child**[1] when you have to **break**[17] a **promise**[3] to be **home**[2] or **giving**[12] a **young**[1] **child**[1] a beeper to make him **feel**[23] more secure when **left**[5] alone.

Fig. 5. Portions of an article about virtual parenting.

TABLE 1
Some Lexical Chains from the Virtual Parenting Article

C	Word	Syn	C	Word	Syn	C	Word	Syn
1	working (5)	40755		expert (1)	59108	12	giving (1)	19911
	ground (1)	58279		mark (1)	60270		pushing (1)	20001
	field (1)	57992		worker (1)	59145		push (1)	20001
	antarctica (1)	58519		speaker (1)	63258		high-tech (2)	19957
	michigan (1)	57513		advertiser (1)	59643	19	planning (1)	23089
	feed (1)	53429		entrepreneur (1)	60889		arranging (1)	23127
	chain (1)	57822		engineer (1)	59101	21	good_night (1)	48074
	hazard (1)	77281		sitter (1)	59827		wish (1)	48061
	risk (1)	77281		consultant (2)	59644	22	phone (2)	40017
	young (2)	24623		management_consultant (1)	61903		cellular_phone (1)	33808
	need (1)	58548		man (1)	61902		fax (2)	35302
	parent (7)	62334		flight_attendant (1)	63356		gear (1)	32030
	kid (3)	60256	4	folk (1)	54362		joint (2)	36574
	child (1)	60256		family (4)	54362		junction (1)	36604
	baby (1)	59820	10	management (2)	55578		network (1)	37247
	wife (1)	63852		professor (1)	62638		system (2)	32196
	adult (1)	59073		conference (1)	55372		audiotape (1)	39983
	traveller (3)	59140		meeting (1)	55371		gadget (1)	32428
	substitute (1)	63327		school (1)	55261	23	feel (1)	22808
	backup (1)	63327		university (1)	55299		kissing (1)	22806
	computer (1)	60118		company (1)	54918			

TABLE 2
Description of the Paragraphs of the Virtual Parenting Article

Par	Chains	Topic
1	1, 3, 4, 6, 8	Introduction of the term *virtual parenting*.
2	1, 2, 3, 6, 7, 12, 13, 16, 17, 21, 22, 23	A definition of virtual parenting — parents using new communication technologies to keep in touch with their kids.
3	1, 3, 4, 9, 10, 12, 13, 19, 20, 22	How businesses are trying to cash in on the trend.
4	1, 3, 4, 8, 10, 12, 15, 18, 21 22	The trend is meeting the need of parents.
5	1, 3, 13, 19, 22	An example: live video over the Internet.
6	1, 3, 10, 13, 14	More examples: using email or recorded videos to keep in touch.
7	1, 3, 4, 9, 11, 13, 17, 22, 24, 25	Advice from communication companies: attend missed Little League games by cellular phone.
8	1, 2, 3, 5, 10, 12, 17, 22, 23, 24	More advice for parents: phone or fax your child when you're traveling.
9	1, 3, 6, 12, 22	A warning from the man who coined the term virtual parenting.
10	1, 2, 3, 4, 8, 10, 13, 22	A warning from someone who designed a system allowing parents to check up on their kids.
11	1, 3, 8	Conclusion: find the middle ground.

4.1 Analyzing the Lexical Chains

We begin our analysis of a document's structure by determining how "important" each chain is to each paragraph in the document. By making this determination, we will be able to link together paragraphs that share sets of important chains. We judge the importance of a chain to a particular paragraph by calculating the fraction of the content words of the paragraph that are in that chain. We refer to this fraction as the *density* of that chain in that paragraph. The density of chain c in paragraph p, $d_{c,p}$, is defined as:

$$d_{c,p} = \frac{w_{c,p}}{w_p},$$

where $w_{c,p}$ is the number of words from chain c that appear in paragraph p and w_p is the number of content words (i.e., those words that are not stop words) in p. For example, if we consider paragraph 1 of our virtual parenting article, we see that there are two words from chain 1. We also note that there are 14 content words in the paragraph. So, in this case, the density of chain 1 in paragraph 1, $d_{1,1}$ is:

$$d_{1,1} = \frac{2}{14} = 0.14.$$

Similarly, we find that $d_{4,1} = 0.07$, and so on. The result of these calculations is that each paragraph in the document will have associated with it a vector of chain densities. Each of these *chain density vectors* will contain an element for each of the chains in the document. Table 3 shows some of the chain density vectors computed for the virtual parenting article.

TABLE 3
Some Chain Density Vectors for the Virtual Parenting Article

Chain	Paragraph										
	1	2	3	4	5	6	7	8	9	10	11
1	0.14	0.19	0.07	0.16	0.28	0.18	0.10	0.25	0.24	0.13	0.33
2		0.02						0.04		0.03	
5								0.04			
8	0.07			0.11						0.03	0.11

4.2 Computing Paragraph Similarity

As we said earlier, the parts of a document that are about the same thing, and therefore related, will tend to contain the same lexical chains. Given the chain density vectors that we computed above, we can compute the similarity between the paragraphs of the document simply by computing the similarity between the chain density vectors representing them. Computing the similarity of all pairs of chain density vectors gives us a symmetric $p \times p$ matrix of similarities, where p is the number of paragraphs in the document.

We can compute these similarities using any one of a number of similarity coefficients that have appeared in the IR literature throughout the years (see [15] for a good discussion of the alternatives available). We rely on the Dice association or Euclidean distance coefficients. If we consider paragraphs 1 and 2 of our example document, then we can compute the similarity of these two paragraphs, $s_{1,2}$, as 0.70, using the Dice coefficient.

4.3 Deciding on the Links

The next step is to decide which paragraphs should be linked on the basis of the similarities computed in the previous step. We make this decision by looking at how the similarity of two paragraphs compares to the mean paragraph similarity across the entire document. Each similarity between two paragraphs i and j, $s_{i,j}$, is converted to a z-score, $z_{i,j}$. That is, each similarity is converted to a measure indicating how many standard deviations away from the average paragraph similarity it is. If two paragraphs are more similar than a threshold given in terms of a number of standard deviations, then a link is placed between them. The result is a symmetric adjacency matrix where a 1 indicates that a link should be placed between two paragraphs.

This z-score metric of similarity is meant to capture our intuition that we want to link paragraphs that are "very similar." The problem is that determining just how similar two paragraphs are will depend on the context in which they occur. Documents with a lot of large chains spread throughout them will tend to display higher interparagraph similarity scores. If we set a simple threshold to determine which paragraphs to link, then in cases such as this we will tend to link almost all pairs of paragraphs. This is clearly not the correct thing to do as this would severely disrupt the reader. What we would like to do is to link only those paragraphs whose similarity significantly deviates from the average. The z-score measure that we have proposed is a traditional method for determining how much a single number stands out from the mean.

Continuing with our example, consider $s_{1,2} = 0.70$. If we know that the mean paragraph similarity is 0.72 and that the standard deviation in paragraph similarity is 0.12, then we can compute $z_{1,2}$ to be -0.17. So, $s_{1,2}$ is 0.17 standard deviations closer to 0 than the mean. If we are using a threshold of 1.0, paragraphs 1 and 2 will not be linked since, in this case, $z_{1,2}$ would have to be greater than 1.0 (since higher scores are better for the Dice coefficient.) If, on the other hand, we consider $s_{2,5} = 0.88$, then we would have $z_{2,5} = 1.33$ and, for a threshold of 1.0, we would link paragraphs 2 and 5.

The result of computing these z-scores is a symmetric adjacency matrix that we can visualize as a set of links between the paragraphs (see Fig. 6). This set of links shows exactly the kind of connections we wanted for this document. The second paragraph (the definition) is linked to paragraphs 5 (an example), 8 (advice), and 9 (a warning). In addition, paragraph 5 is linked to paragraphs 8 and 9.

4.4 Examining a Connection

At this point we should step back and look at the relations between the words in the linked paragraphs. For example, consider the link that was built between paragraphs 2 and 8. This connection was built on the strength of the seven chains that they have in common: chains 1, 2, 3, 12, 17, 22, and 23. Fig. 7 shows these two paragraphs with only words from these chains highlighted in bold. Terms which are repeated across the two paragraphs are shown in italics. Thus, bold italic terms are both in one of these chains and repeated.

Although there is a small amount of term-repetition between these paragraphs (e.g., *business* and *phone* are repeated), standard IR methods would not have enough data available to make the connection. The lexical chains, on the other hand, connect together synonyms such as *kid* and *child*. More-distant connections are also made between the paragraphs, such as the fact that phones, cellular phones, and faxes are all communication media, or the fact that there is a relation between the words *parent* and *child* (i.e., parents have children.) This extra information allows the linker to make the connection between these two paragraphs and build a link between them.

At this point, we will note that the process of lexical chaining is not perfect and, so, we must accept some errors (or at least bad decisions) for the benefits that we receive. In our sample article, for example, chain 1 is a conglomeration of words that would have better been separated into different chains. This is a side effect of the

BNSDOCID: <XP_____2317545A_I_>

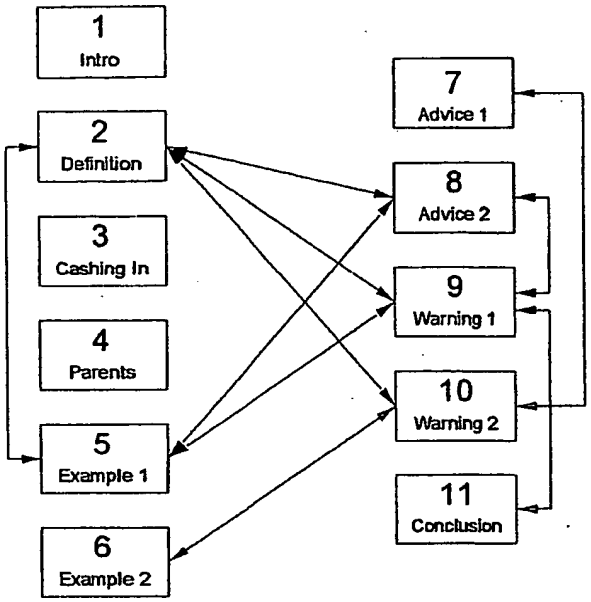

Fig. 6. Links between paragraphs for the virtual parenting article.

current implementation of the lexical chainer, but even with these difficulties, we are able to perform useful tasks.

4.5 Generating a Hypertext Representation

Once the intradocument hypertext links have been decided on, a representation of the hypertext that can be used for browsing needs to be produced. We have decided to use HTML as our hypertext representation since it is an open standard and relatively easy to use. This is not to say that HTML is the only possible (or even the best) representation, and we have taken care to ensure that the hypertexts that our method produces will be usable in other hypertext systems.

In the current system, there are two ways to output the HTML representation of a document. The first simply displays all of the links that were computed during the last stage of the process described above. The second is more complicated, showing only some of the links. The idea is that links between physically adjacent paragraphs should be omitted so that they do not clutter the hypertext, making it more difficult to use.

5 Linking Between Documents

While it is useful to be able to build links within documents, for a large scale hypertext, links also need to be placed between documents. You will recall from Section 3 that the output of the lexical chainer is a list of chains, each chain consisting of one or more words. Each word in a chain has associated with it one or more synsets. These synsets indicate the sense of the word as it is being used in this chain. An example of the kind of output produced by the chainer is shown in Table 4, which shows the chains extracted from an article about cuts in staff at children's aid societies due to a reduction in Canadian provincial grants [16]. As before, the numbers in parentheses show the number of occurrences of a particular word. Table 5 shows another set of chains, this time from an article describing the changes in child-protection agencies, due in part to budget cuts [17].

It seems quite clear that these two documents are related, and that we would like to place a link from one to the other. It is also clear that the words in these two

Although no one is **pushing**[12] virtual-reality headgear as a **substitute**[1] for **parents**[1], many technical ad campaigns are promoting *cellular phones*[22], **faxes**[22], **computers**[1] and pagers to **working**[1] **parents**[1] as a way of bridging **separations**[17] from their *kids*[1]. A recent promotion by A T & T and Residence[2] Inns in the United States, for **example**[3], suggests that *business*[3] *travelers*[1] with **young**[1] children use **video**[3] and audiotapes[22], **voice**[3] **mail**[3], videophones and E-mail to **stay**[3] connected, including **kissing**[20] the **kids**[1] good night by **phone**[22].

More **advice**[3] from advertisers[1]: *Business*[3] *travelers*[1] can dine with their *kids*[1] by speaker[1]-*phone* or "tuck them in" by cordless *phone*[22]. Separately, a management newsletter recommends faxing your **child**[1] when you have to break[17] a **promise**[3] to be **home**[2] or **giving**[12] a **young**[1] **child**[1] a beeper to make him feel[23] more secure when left alone.

Fig. 7. Paragraphs 2 and 8 of the virtual parenting article.

BNSDOCID: <XP 2317545A >

TABLE 4
Lexical Chains from an Article about Cuts in Children's Aid Societies

C	Word	Syn	C	Word	Syn	C	Word	Syn
3	society (7)	54351		annual (1)	64656		care (1)	22204
	group (1)	19698	5	ontario (1)	56918		social_work (1)	24180
	mother (1)	62088		canadian (1)	58424		slowdown (1)	23640
	parent (4)	62334			59296		abuse (3)	21214
	kid (1)	60256		burlington (1)	57612		child_abuse (1)	21215
	recruit (1)	62769		union (3)	57424		neglect (1)	21235
	employee (2)	60862	10	saying (1)	50294	28	living (1)	75629
	worker (2)	59145		interview (2)	50268		standing (1)	75573
	computer (1)	60118	27	try (1)	22561		complaint (1)	76270
	teen-ager (2)	59638		seeking (1)	22571		agency (1)	75786
	provincial (3)	62386		acting (1)	21759		stress (1)	76799
	face (1)	59111		services (1)	21922			76906
	spokesman (1)	63287		work (3)	21919	32	executive_director (2)	60922
	insolvent (1)	59869		risk (2)	22613		manager (1)	59634

TABLE 5
Lexical Chains from a Related Article

C	Word	Syn	C	Word	Syn	C	Word	Syn
2	wit (1)	48647		guardian (1)	59099			24236
	play (1)	48668		official (1)	62223		making (1)	23076
	abuse (4)	48430		worker (1)	59145		calling (1)	21911
	cut (4)	48431		neighbour (1)	62152		services (2)	21922
	criticism (1)	48406		youngster (1)	60255		prevention (1)	23683
	recommendation (1)	48310		kid (2)	60255		supply (1)	23596
	case (1)	48682		natural (1)	62139		providing (3)	23596
	problem (1)	48680		lawyer (2)	61725		maltreatment (2)	21214
	question (3)	48679		professional (1)	62636		child_abuse (2)	21215
3	child (10)	60256		prostitute (1)	62660		investigation (1)	22142
	parent (9)	62334		provincial (2)	62386		research (1)	22143
	mother (3)	62088		welfare_worker (1)	63220		investigating (1)	22142
	daughter (1)	60587		lorelei (1)	61833		work (1)	21885
	foster_home (1)	54374		god (1)	58615		aid (9)	22204
	society (5)	54351	4	protection (2)	22672		social_work (1)	24180
	at_home (1)	55170		care (5)	22721		risk (1)	22613
	social (1)	55184		preservation (2)	22676		dispute (1)	24051
	function (1)	55154		judgment (1)	22881		intervention (1)	24317
	expert (3)	59108		act (1)	19697		fail (1)	19811
	human (1)	19677		behaviour (1)	24235			

documents display both of the linguistic factors that affect IR performance, namely synonymy and polysemy. For example, chain 27 in Table 4 contains the word *abuse*, while chain 4 in Table 5 contains the synonym *maltreatment*. Similarly, the first set of chains includes the word *kid*, while the second contains *child*. The word *abuse* in the first article has been disambiguated by the lexical chainer into the "cruel or inhuman treatment" sense, as has the word *maltreatment* from the second article. We once again note that the lexical chaining process is not perfect: For example, both texts contain the word *abuse*, but it has been disambiguated into different senses.

Although the documents share a large number of words, by missing the synonyms or by making incorrect (or no) judgments about different senses, a traditional IR system might miss the relation between these documents or rank them as less related than they really are. Aside from the problems of synonymy and polysemy, we can see that there are also more-distant relations between the words of these two documents. For example, the first set of chains contains the word *maltreatment*, while the second set contains the related term *child abuse* (a kind of maltreatment).

Our aim is to build hypertext links between documents that will account for the fact that two documents that are about the same thing will tend to use similar (although not necessarily the same) words. These *interdocument* links can be built by determining how links could be built between the words of the chains from the two documents. By using the lexical chains extracted from the documents, rather than just the words, we can account for the problems of synonymy and polysemy, and we can take into account some of the more-distant relations between words.

BNSDOCID: <XP_____2317545A__I_>

5.1 Comparing Chains Across Documents

We want to make the comparison between two documents on the basis of the lexical chains that have been extracted from the documents. This comparison could be seen as the same kind of operation that was done during the initial chaining of both documents, that is, this comparison is a kind of "cross-document" chaining. The main difference between chaining within a document and cross-document chaining is that, in cross-document chaining, we want to restrict the chaining algorithm so that only extra strong and strong relations are allowed. We enforce such a restriction for two reasons. First, allowing regular relations between words will introduce too many spurious connections. We allow it at the document level so that intradocument links can be built more easily. Second, finding regular relations is the most time-consuming part of the lexical chaining process and, so, it cannot be done in real-time, which would be necessary for an on-line system.

5.2 An Initial Approach

If we wish to link two documents using their lexical chains, taking into consideration the above criteria, then there is a straightforward solution. Given two sets of chains, we can simply determine the number of strong and extra-strong links between the synsets that appear in the chains extracted from the two documents. Once we have determined this number, we can decide whether they should be related. The main strength of this algorithm is its simplicity. It is easy to implement and understand. It also has the desirable property that documents that contain the same term can only be related when the two words share the same synset (i.e., when the words are used in the same sense).

Unfortunately, this approach also has some rather debilitating weaknesses. Due to the hierarchical structure of WordNet, it is very easy to find documents that have a large number of related words, even when the documents are completely unrelated. When a word in a chain is in synsets that are near the top of WordNet's hierarchy, there are a large number of synsets that are a single IS-A or INCLUDES link away. Very general words like *human* can be linked to a large number of other words. This is especially a problem when the documents in question are long since there is more opportunity for such connections.

The other main weakness is that this approach is extremely time consuming. Our calculations indicate that, using this method, it would take approximately six years to determine all possible interdocument links for a database consisting of one year's articles from a typical newspaper. If we attempt to do this in real-time and simply search through a year of documents to find links from a particular document, we can reduce the time to approximately one hour. Unfortunately, this is still unacceptable.

The problem is that there is no straightforward, global description for a document, so each set of chains must be treated as a special case. In traditional vector space IR systems, the term weight vector provides such a global description. This vector is the same length for each document, and a particular element of the vector is used for the weight of a particular term in every document. Lexical chaining, on the other hand, is more fluid. It is highly unlikely that two documents will contain the same set of lexical chains. In the vector space model, it is a simple decision to say whether two documents have a term in common; all that is required is to check the term weight vector. Discovering related documents is as simple as taking the dot product of two vectors. It is quite difficult to say that two documents have related chains since it is necessary to try to relate each of the words in the two chains of interest.

In order to build a system that is reasonably efficient, we need to devise a simple, global representation for the lexical chains which retains the properties of disambiguation and linking-by-relation as the method described above, while at the same time dealing with the problem of spurious links.

5.3 Synset Weight Vectors

In fact, such a simple, global representation is reasonably close at hand. In the vector space model for information retrieval, documents are represented by weighted term vectors. The weight of a particular term in a particular document is not based solely on the frequency of that term in the document, but also on how frequently that term appears throughout the entire database of documents. The terms that are the most heavily weighted in a document are the ones that appear frequently in that document but infrequently in the entire database.

We propose that a document be represented by two vectors. Each vector will have an element for each synset in WordNet. An element in the first vector will contain the weight of that particular synset in the document. An element in the second vector will contain the weight of that particular synset when it is one link away from a synset that appears in the lexical chains of a document. We will call these vectors the *member* and *linked weighted synset vectors*, or simply the *member* and *linked vectors*, respectively.

The equation from Salton and Allan [18] used to compute term weights serves equally well when computing weights for synsets:

$$w_{ik} = \frac{sf_{ik} \cdot \log(N/n_k)}{\sqrt{\sum_{j=1}^{t} (sf_{ij})^2 \cdot (\log(N/n_j))^2}}.$$

Here, w_{ik} is the weight of *synset* k in document i, sf_{ik} is the frequency of synset k in document i, n_k is the number of documents that contain synset k, and N is the number of documents in the entire collection.

In our case, rather than calculate a single set of weights incorporating the frequencies of both member and linked synsets, the weights are calculated independently for the member and linked vectors. We do this because the linked vectors introduce a large number of synsets that do not necessarily appear in the original chains of a document and should therefore not influence the frequency counts of the member synsets. Thus, we make a distinction between strong relations between documents that occur due to synonymy and ones that occur due to IS-A or INCLUDES relations. In the former case, such relations will be part of the member vector, while in the later case, they will be found in the linked vector.

These synset weight vectors can be seen as a *conceptual* or *semantic* representation of the content of a document, as

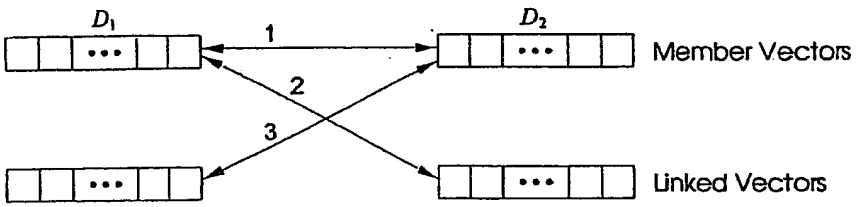

Fig. 8. Computing chain similarity.

opposed to the traditional IR method of representing a document by the words that it contains. This representation also addresses both synonymy and polysemy. Synonymy is taken care of by virtue of the fact that all of the synonyms for a word will be collected in the same synset and, therefore, represented in the same element of the synset vectors. Because of the disambiguation performed by the lexical chainer, a word will be represented only by synsets (i.e., senses) that are appropriate in the context of the document. Only these synsets will appear in the weighted synset vectors, solving (to some extent) the problem of polysemy.

We can then compute the relatedness of two documents D_1 and D_2 by measuring three similarities (shown by the lines in Fig. 8):

1. the similarity of the member vectors of D_1 and D_2;
2. the similarity of the member vector of D_1 and linked vector of D_2; and
3. the similarity of the linked vector of D_1 and the member vector of D_2.

Clearly, the first similarity measure (which we call the *member-member* similarity) is the most important, as it will capture extra strong (i.e., term repetition) relations as well as strong relations between synonymous words. The last two measures (called the *member-linked* similarities) are less important as they capture strong relations that occur between synsets that are one link away from each other.

Once we have built a set of synset weight vectors for a collection of documents, the process of building links between documents is relatively simple. Given a document that we wish to build links from, we can compute the similarity between the document's synset weight vectors and the vectors of all other documents. If the member-member similarity of two documents is higher than a given threshold, then we can calculate the two member-linked similarities and place a link between the two documents. We can rank the links using the sum of the three document similarities that we compute. Our work shows that a threshold of 0.2 will include most related documents while excluding many unrelated documents.

By using such a strenuous threshold, we enforce our constraint that there must be multiple connections between the chains of the documents. This is almost exactly the methodology used in vector space IR systems with the difference being that, for each pair of documents, we are calculating three separate similarity measures. By using the sum of the three similarities as our ranking criterion, we are taking full account of not only the terms and synonyms that the documents have in common, but also how many more distantly related terms they share. The sum of the three similarities can lie, theoretically, anywhere between 0 and 3. In practice, the sum is usually less than 1. For example, the average sum of the three similarities when running the vectors of a single document against 5,592 other documents is 0.039.

As a side-effect of representing documents by the synsets that they contain, we reduce the size of the vectors needed to represent each document. For a database of four months of the *Globe and Mail* (a major Canadian newspaper), we find that there are 31,360 distinct synsets in the member vectors and 46,612 distinct synsets in the linked vectors. Thus, the combined size of the two vectors necessary to represent a document (77,962) is substantially smaller than the more than 108,000 unique terms that Forsyth [19] says we can expect. This reduction in dimensionality is similar to the reduction that we see in Latent Semantic Indexing [20], although their reduction is even more substantial than ours (from 108,000 terms to 200 factors).

5.3.1 How Related Words Affect Linking

Now that we have settled on a method for building interdocument links, we can see how the two sets of chains shown in Table 4 and Table 5 are handled. Table 6 and Table 7 give information about the member and linked vectors that represent these two articles.

If we are using a linking threshold of 0.2, then we will place a link between these documents. The sum of the similarities for the two documents is 0.399. Approximately

TABLE 6
Lengths of the Vectors in the Example Articles

Document	Vector	Length
1	Member	128
1	Linked	574
2	Member	215
2	Linked	1481

TABLE 7
Similarities of the Vectors in the Example Documents

	Document 1	
Document 2	Member	Linked
Member	0.224	0.096
Linked	0.079	—

TABLE 8
Questions Used in Evaluation of Linking Methodology

Number	Answers	Question
Test	N/A	List the names of as many premiers of Canadian provinces as you can find. Be sure to include the name of the province.
1	61	List all the drug brand names that you can find, if you can also list the name of a generic substitute for the drug *or* the chemical name of the drug.
2	56	List the names of as many people as you can find that are identified as "terrorists". You should *not* include the names of terrorist groups.
3	34	List the names of biotechnology companies that have participated in mergers or joint ventures. You should list the names of all participants in the merger or joint venture.

23 percent of the member-member similarity of these documents is accounted for by synsets from which the documents do not share exactly the same words. This proportion of the similarity is sufficiently large that, if it were removed, the member-member similarity of these documents would fall below the linking threshold that we had set.

6 Evaluating the Linking Methodology

Clearly, methodologies such as the one that we have presented in the previous two sections require evaluation. In this section, we will describe the design and results of a study that was undertaken to test our linking methodology.

We will not attempt to answer the question of whether browsing is a useful way of performing IR tasks, as it seems clear that browsing is a viable and necessary component of any IR system (see, for example, [21]). Rather, we will be asking the question: Is our hypertext linking methodology superior to other methodologies that have been proposed (e.g., that of Allan [7])? The obvious way to answer the question is to test whether the links generated by our methodology will lead to better performance when they are used in the context of an appropriate IR task.

The null hypothesis for our tests is simply that there is no significant difference between the hypertext links generated by our method and those generated by another methodology—one could perform IR tasks equally well using either kind of links. Our research hypothesis is that our method provides a significant improvement, because it is based on semantic similarity of concepts rather than strict term repetition.

6.1 Experimental Design

6.1.1 The Task

We selected a questioning-answering task for our study. We made this choice because it appears that this kind of task is well-suited to the browsing methodology that hypertext links are meant to support. This kind of task is also useful because it can be performed easily using only hypertext browsing. This is necessary because, in the interface used for our experiment, no query engine was provided for the subjects.

It may be argued that the restriction to strict hypertext browsing creates an unnatural setting for the study and that, in any real system, users would at least be able to perform a keyword search. This may be true, but if we had included a query engine, then it is possible that any results that we obtained would pertain more to the use of queries rather than browsing or to how well users can form queries. By making the restriction, we tested just the hypothesis in which we were interested: Is a semantically based approach to hypertext link generation better than a strict term-repetition approach? If we can make a determination one way or the other, then we will be able to draw conclusions about how hypertext links should be built in a system that provides both querying and browsing.

6.1.2 The Questions and the Database

The most difficult part of performing an evaluation of any IR or hypertext system is developing reasonable questions and then determining which documents from the test database contain the answers. Several test collections have been developed over the years that can be used by anyone who wishes to compare the performance of his or her IR system to others. The most recent, and certainly the largest, of these collections is the TREC collection. We used the "Narrative" section of three TREC topics as the basis of the test questions shown in Table 8.

There were approximately 1,996 documents that were relevant to the topics from which these questions were created. We read these documents and prepared lists of answers for the questions. Our test database consisted of these documents combined randomly with approximately 29,000 other documents selected randomly from the TREC corpus. The combination of these documents provided us with a database that was large enough for a reasonable evaluation and yet small enough to be easily manageable. As most of the documents in the database were newspaper or newswire articles, the test database was presented to the users as a "database of newspaper articles."

6.1.3 Whose Links to Use?

We considered two possible methods for generating interdocument hypertext links. The first is our method, described earlier. The second method uses a vector space IR system called Managing Gigabytes (MG) [22] to generate

BNSDOCID: <XP_____2317545A_I_>

links by calculating document similarity. We used the MG system to generate links in a way very similar to that presented in Allan [7].

Links from a source document were built by passing the entire text of the source document to the MG system as a "query." MG builds the term vector representing this query after removing stop words and stemming the words in the query. This query vector was compared against the document vectors stored in the MG database, and the top 150 related documents were returned and used as the targets of the interdocument hypertext links. The MG system provided most of the same capabilities as the SMART system used by Allan. We used the MG system because it was much more easily integrated into our other software. For simplicity's sake, we will call the links generated by our technique *HT links* and the links generated by the MG system *MG links*.

At this point, we considered two approaches to testing the effectiveness of these two sets of links. The first was to set two experimental conditions: one using HT links and the other using MG links. This is a very typical experimental strategy, and certainly viable in this case. The problem was that such a design would have required a large number of subjects to be tested in each condition to ensure that the study was valid.

The second method was, at each stage during a subject's browsing, to combine the sets of links generated by the two methods. This results in a single experimental condition where the system must keep track of how each interdocument link was generated. By using this strategy, the subjects "vote" for the system that they prefer by choosing the links generated by that system. Of course, the subjects are not aware of which system generated the links that they are following—they can only decide to follow a link by considering the article headlines displayed as anchors. We can, however, determine which system they "voted" for by considering their success in answering the questions they were asked. If we can show that their success was greater when they followed more HT links, then we can say that they have "voted" for the superiority of HT links. A similar methodology has been used previously by Nordhausen et al. [23] in their comparison of human and machine-generated hypertext links.

The two sets of interdocument links can be combined by simply taking the unique links from each set, that is, the links that appear in only one of the sets of links. Of course, we would expect the two methods to have many links in common, but it is difficult to tell how these links should be counted in the "voting" procedure. By leaving them out, we test the differences between the methods rather than their similarities. Of course, by excluding the links that the methods agree on we are reducing the ability of the subjects to find answers to the questions that we have posed for them. This appears to be a necessary difficulty of this method and, as we shall see, the number of correct answers that the subjects found was generally quite low, but it was nonetheless sufficient to compare the two methodologies.

The intradocument links that were presented to the users were generated by the methodology described in Section 4. Because there was no other method for generating these links, the subjects were presented only with links generated by our method.

6.1.4 The Evaluation System

The evaluation system used a front-end written in Java combined with a back-end written in C++. Although we have discussed the use of our system over the World-Wide Web, we found it necessary to use a non-Web-based system to perform the evaluation. This was mostly due to the difficulty in obtaining sufficient logging information (e.g., What links were followed?) from a Web browser.

The interface of the system was quite straightforward. It consisted of a single screen similar to the one shown in Fig. 9. The main part of the screen showed the text of a single document. The subjects could navigate through the document by using the intradocument links, the scroll bar, or the page up and down keys. The buttons to the left of the document could be used for navigating through the set of documents that had been visited (the *Previous Article* and *Next Article* buttons) or navigating within a document (the *Back* button would return to the point from which an intradocument link was taken).

At the bottom of the screen was a list of the documents from the database that were related to the document displayed. The anchor text for these links was the headline of the article that the user would jump to when the link was clicked on. In order to leverage the subjects' experience with Web browsers such as Netscape Navigator, all hypertext links were shown in blue, while all regular text appeared in black. To ease navigation difficulties (i.e., "Have I been here before?"), links that had already been traversed (both intradocument and interdocument) were shown in magenta.

6.1.5 Performing Searches

To begin, subjects were given a set of instructions on using the system and were allowed to ask questions about the interface. The subjects were all provided with the "test" question and allowed 5 minutes to become familiar with the properties of the system. Once comfortable, the subjects were given the rest of the questions one by one. The time for each question was limited to 15 minutes so that subjects would not spend inordinate amounts of time on one query and then give the others short shrift. The order in which questions were given was varied among the six possible orders across all of the subjects who performed the task.

Each search began on a "starter" page that contained the text of the appropriate TREC topic as the "document" and the list of documents related to the topic shown (this was computed by using the text of the topic as the initial "query" to the database). Subjects were expected to traverse the links, writing down whatever answers they could find. As the subjects browsed through the database of documents, the links that they followed within and between documents were automatically logged. In addition, any scrolling motions within a document were recorded (e.g., using the scrollbar or the page up and down keys). When a subject left one document to go to another, the amount of

BNSDOCID: <XP_____2317545A__I_>

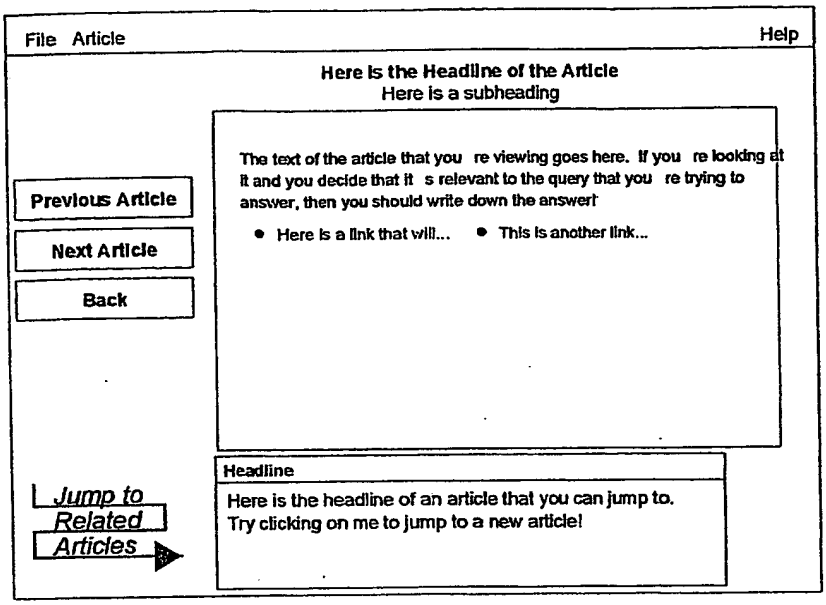

Fig. 9. The interface of the evaluation system.

time spent on the document was recorded. After they had finished answering the questions, the subjects were given a short questionnaire to fill out.

6.2 Analysis

We tested 27 subjects during the course of the evaluation. However, our analysis will only include 23 subjects. Some changes were made after the first day of the evaluation in order to improve the reliability of the Java front-end, resulting in significantly fewer disruptive system crashes. In one case during the first day, the system crashed five times during the course of one 15-minute question. More importantly, the way in which interdocument links were displayed was changed. In addition to the system changes, we corrected a grammatical error in one of the questions and slightly modified the instructions that were provided to the subjects. Because of these changes, we decided that it would be best if the results from the first day were removed from consideration during the analyses, since those subjects were not operating under the same set of experimental conditions as the others.

A summary of the data is shown in Table 9 (full data can be found in [2]). In this table, the variable name L_{MG} refers to the number of MG links followed, L_{HT} refers to the number of HT links followed, L_I refers to the number of intradocument links followed, and *Ans* refers to the number of correct answers found.

TABLE 9
Summary Statistics for Experimental Results

Data	Min	Max	Mean	Std. Dev.
L_{HT}	1	19	6.87	3.44
L_{MG}	0	15	6.32	3.20
L_I	0	23	4.97	5.41
Ans	0	16	4.48	2.98

The number of both interdocument and intradocument links followed was, on average, quite small and variable. As we expected, the number of correct answers found was also low and variable. On average, the subjects showed a slight bias for HT links, choosing 52.1 percent HT links and 47.9 percent MG links. This is interesting, especially in light of the fact that, for all the documents the subjects visited, 50.4 percent of the links available were MG links, while 49.6 percent were HT links. A paired t-test, however, indicates that this difference is not significant.

We can also combine L_{HT} and L_{MG} in a ratio that we will call L_R. Because $L_{MG} = 0$ in some cases, we will define L_R in the following way:

$$L_R = \begin{cases} \frac{L_{HT}}{L_{MG}} \text{ when } L_{MG} > 0 \\ L_{HT} \text{ when } L_{MG} = 0 \end{cases}$$

If $L_R > 1$, then a subject followed more HT links than MG links. An interesting question to ask is: Did subjects with significantly higher values for L_R find more answers, that is, did people who followed more HT links find more answers? With 23 subjects each answering three questions, we have 69 values for L_R. If we sort these values in decreasing order and divide the resulting list at the median, we have two groups with a significant difference in L_R. An unpaired t-test then tells us that the differences in *Ans* should occur by chance with $p < 0.1$. This is certainly unlikely enough that there may be some relationship between the number and kinds of links that a subject followed and his or her success in finding answers to the questions pose. In the following sections, we will explore

TABLE 10
ANOVA Analysis for a Regression Model with an Intercept

Source	SS	df	Mean Square	F	p
Regression	87.13	2	43.56	5.55	0.01
Error	518.09	66	7.85		

TABLE 11
Ninty-Five Percent Confidence Intervals for a Model with an Intercept

Param	Value	Error	t	p	Low	High
Const	2.08	0.98	2.11	0.02	0.11	4.04
L_{HT}	0.33	0.10	3.30	0.00	0.13	0.52
L_{MG}	0.03	0.11	0.24	0.41	−0.19	0.24

this relationship using regression analyses. In fact, there are two cases that we wish to consider. In the first, we look at only the interdocument links that the subjects followed. In the second, we include the intradocument links as well.

6.2.1 Interdocument Links

In the first case, we will consider solely the relationship between the kinds of interdocument links that the subjects used (i.e., HT versus MG links). We can use a multivariate regression model with two independent variables, L_{MG} and L_{HT} relationship between HT links, MG links, and the number of correct answers found. The dependent variable in our analysis is *Ans*, the number of correct answers found by the subject. For each subject, we will have three measurements of the independent and dependent variables corresponding to the three questions that they answered.

Note that we are using the number of correct answers that the subjects found as our dependent variable. It may be argued that a more appropriate measure would be the percentage of the possible answers that they found—essentially the recall of the correct answers. This would be a valid concern for an evaluation in which the subjects were allowed to look for answers until they felt they had found them all. In our task, however, searches were limited to 15 minutes and the speed of the system tended to limit the number of answers that a subject could find. For example, there was no significant difference in the number of answers between questions 1 and 3, even though question 1 has nearly twice as many possible answers as question 3. If we were to use the percentage of correct answers found, then we would artificially lower the subjects' scores.

6.2.1.1 A Standard Regression. Our regression model gives us the following equation for deriving the number of correct answers found from the number of each type of link followed:

$$Ans = 2.08 + 0.33 \cdot L_{HT} + 0.03 \cdot L_{MG} \quad (R^2 = 0.14).$$

So, at least at first glance, it seems that, by following an HT link, a user would derive a greater benefit (in terms of the number of correct answers found) than she would get from traversing an MG link. Unfortunately, the analysis is not that simple. We also need to ask ourselves what the possibility is that the independent variables that we have chosen are actually unrelated to the dependent variable. We can test this hypothesis with an ANOVA analysis of the linear regression to see how much of the difference between the observed and fitted values of *Ans* is attributable to the regression and how much to simple error. The ANOVA table is shown in Table 10.

For the calculated value of F, we can reject the initial hypothesis that L_{MG} and L_{HT} are unrelated to *Ans* with $p < 0.01$. Now, if our dependent variable is related to our independent variables, then we still need to ask what range of values we can reasonably expect the coefficients of our independent variables to take on. Table 11 shows the 95 percent confidence intervals for these coefficients, which provides an estimate of this range.

Here, the column labeled t is the t-score associated with the hypothesis H_0: The coefficient in question is 0. The alternative hypothesis is that the coefficient is greater than 0. The column labeled p is the probability that H_0 is true. For this model, we can safely reject H_0 for the coefficient of L_{HT} with $p < 0.05$. We can also reject H_0 for the constant in our equation. This is surprising, as we told the subjects to record only those answers that they found in the database, and not those that they already knew. In addition, there were no answers on any of the "starter" pages for the questions. So, if a subject followed no links, then they should have been unable to find any answers and *Ans* should therefore have been 0. Interestingly, we cannot reject H_0 for L_{MG}, meaning that the coefficient may be 0.

The columns labeled *Low* and *High* give the endpoints of the 95 percent confidence interval for the values of each of the coefficients. Notice that the confidence intervals for the coefficients of L_{MG} and L_{HT} overlap significantly. This leads us to the conclusion that it is possible that, for this model, the coefficient of L_{MG} may be greater than the coefficient of L_{HT} some of the time if, in fact, the coefficient of L_{MG} is not 0. Thus, for this case, we cannot reject our null hypothesis that the number of answers that a user will find does not depend on which kind of links that they follow.

6.2.1.2 Removing the Constant. In the previous model, we noted that we could not necessarily say that the constant term was 0, even though this was to be expected. Also, we were unable to say that the coefficient of L_{MG} was greater than 0. This would seem to be a useful result for us since we could say that following MG links has no benefit. However, as we are proposing an alternative method, we feel that we should give the MG method of generating links the benefit of the doubt in this case. So, we propose another regression model in which we ensure that the fitted value of the constant is its theoretical value of 0. This model results in the equation:

$$Ans = 0.46 \cdot L_{HT} + 0.17 \cdot L_{MG} \quad (R^2 = 0.09),$$

which shows a smaller benefit than the previous model for the selection of an HT link over an MG link. An ANOVA analysis of this model shows that our dependent variables are related to our independent variable and that, with

TABLE 12
Ninty-Five Percent Confidence Intervals for a Model without an Intercept

Param	Value	Error	t	p	Low	High
L_{HT}	0.46	0.08	5.96	0.00	0.31	0.62
L_{MG}	0.17	0.08	2.01	0.02	0.00	0.34

$p \leq 0.05$, we can safely assume that the number of links followed is related to the number of answers found.

The 95 percent confidence intervals for the model coefficients are shown in Table 12. Notice that the standard errors for the coefficients have dropped when compared to the ones in Table 11 and that we can now safely reject the hypothesis that the coefficients of the model parameters are 0 for all of the coefficients. Unfortunately, there is still an overlap in the confidence intervals for the coefficients of L_{HT} and L_{MG}, so we cannot reject our null hypothesis in this case. We do note, however, that the overlap is relatively small. By inspection, we find that the confidence intervals begin overlapping at approximately the 92.5 percent level.

6.2.2 A Two-Dimensional Model

Rather than casting our data as a three-dimensional regression problem, we could instead consider the question of how L_R (the ratio of HT links to MG links) and *Ans* (the number of correct answers) are related. If we can show that the regression line for these two variables has positive slope, then we will know that increasing the number of HT links that a user takes will increase his or her number of correct answers.

This model gives us the following equation for the regression line:

$$Ans = 3.65 + 0.56 \cdot L_R \quad (R^2 = 0.05).$$

Fig. 10 shows a scatter plot of the values and the regression line. Notice that the intercept is quite high, almost at the average for the data that we collected. An ANOVA analysis similar to those above, however, shows us that L_R is related to *Ans* with $p < 0.07$. Table 13 shows the 95 percent confidence intervals for the parameters of this model. From this table, we see that we can reject the hypothesis that the coefficient of L_R is 0 with $p < 0.05$. We note, however, that a very small portion of the 95 percent confidence interval is negative, indicating that, some of the time, we could expect a greater benefit from following MG links rather than HT links.

6.2.2.1 **Data by Experience.** We can also ask how a subject's success is affected by their degree of previous experience in using hypertext. The questionnaire given to the subjects asked how often they browse the Web. We can take their answers to this as an indication of their experience using hypertext. We divide the subjects into two groups. The first group, which we will call the *Low Web* group, indicated that they use the Web less than three times a week. The second group, the *High Web* group, indicated that they use the Web three or more times a week. An unpaired t-test shows that the High Web group (12 subjects) chose significantly more ($p < 0.01$) interdocument links than the Low Web group (11 subjects). This difference indicates that these subjects are probably more comfortable in a hypertext environment than the other subjects, and adapted more quickly to the interface used for the task.

When we look at the numbers of each kind of hypertext links followed by each group, we see that the High Web group chose significantly more HT links than the Low Web group ($p < 0.01$). There was no significant difference in the

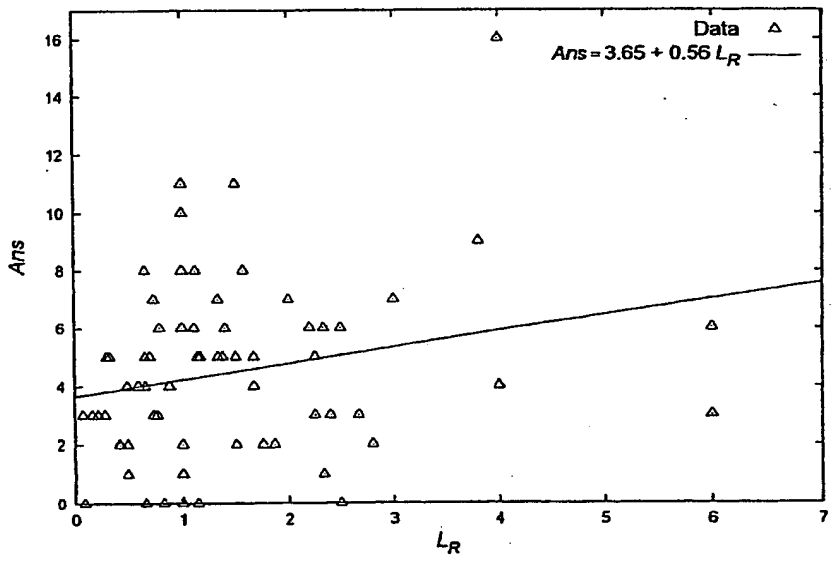

Fig. 10. Data and regression line for all questions.

BNSDOCID: <XP_____2317545A_I_>

TABLE 13
Ninty-Five Percent Confidence Intervals
for a Two-Dimensional Model of All Data

Param	Value	Error	t	p	Low	High
Const	3.65	0.56	6.52	0.00	2.53	4.77
L_R	0.56	0.30	1.90	0.03	−0.03	1.16

TABLE 14
Ninty-Five Percent Confidence Intervals
for Coefficients in a Model Using Viewed Answers

Param	Value	Error	t	p	Low	High
L_{HT}	0.70	0.11	6.57	0.00	0.49	0.92
L_{MG}	0.26	0.12	2.28	0.01	0.03	0.50

number of MG links chosen by the two groups. Within each group, we find that the High Web group chose significantly ($p < 0.05$) more HT links than MG links, while there was no such significant difference in the Low Web group. There is also a significant difference ($p < 0.01$) in the number of answers found by the two groups, with the High Web group finding more correct answers.

If we consider transforming our ratio measure by taking its inverse, $\frac{1}{L_R}$, then we see a significant ($p < 0.05$) difference in the ratios between the High and Low Web groups. Thus, we can see a set of subjects (the High Web group) who found significantly more answers and followed significantly more HT links, indicating the advantage of HT links over MG links.

As with our other data sets, we can build two-dimensional regression models for each of these groups. The models for these groups produce the following equations:

$$\text{Low Web}: Ans = 2.56 + 0.73 \cdot L_R \ (R^2 = 0.12),$$

and

$$\text{High Web}: Ans = 4.92 + 0.25 \cdot L_R \ (R^2 = 0.01).$$

Although only the model for the Low Web group is significant, we see that the slope of the regression line for the Low Web group is steeper than that for the High Web group, indicating that the Low Web group benefited more from following HT links than did the High Web group.

6.2.3 Viewed Answers

In the analyses that we've performed to this point, we have been using the number of correct answers that the subjects provided as our dependent variable. We have also mentioned that the reason we are using this dependent variable is that the subjects were limited in the amount of time that they could spend on each search. We can mitigate this effect by introducing a new dependent variable, Ans_V, or the number of *viewed answers*.

The number of viewed answers for a particular question is simply the number of answers that were contained in documents that a subject visited while attempting to answer a question. These answers need not have been written down. We are merely saying that, given more time, the subjects might have been able to read the document more fully and find these answers. This idea is analogous to the use of *judged* and *viewed recall* by Golovchinsky [24] in his studies.

For the data collected from our study, a paired t-test indicates that there is a significant difference ($p \approx 0$) between Ans_V and Ans, so we could investigate a two-dimensional regression model using Ans_V as the dependent measure; however, such a model is not significant. We must then return to a three-dimensional model incorporating separate terms for L_{MG} and L_{HT}. Such a model is highly significant and gives us the following equation:

$$Ans_V = 0.70 \cdot L_{HT} + 0.26 \cdot L_{MG} \ (R^2 = 0.22),$$

which shows a greater benefit for HT links over MG links. The 95 percent confidence intervals for this model, however, do show a very small overlap (less than 1 percent of the interval for L_{HT}) between the coefficients of L_{MG} and L_{HT}, as we see in Table 14. This overlap precludes us from claiming significance for this result.

6.2.4 Interdocument and Intradocument Links

While we're primarily interested in how well our interdocument linking works compared to other methods, we are also interested in seeing how the use of intradocument links affected the number of correct answers that a user found. We can begin answering this by proposing a regression model in which the independent variables are L_{MG}, L_{HT}, and L_I and the dependent variable is Ans. For simplicity's sake, we will show only the model in which the constant has been fixed at 0.

This model gives us the following relationship between the three types of links and the number of correct answers:

$$Ans = 0.44 \cdot L_{HT} + 0.15 \cdot L_{MG} + 0.06 \cdot L_I \ (R^2 = 0.10).$$

As with the model discussed above, there is still a greater benefit in selecting an HT link over an MG link. The coefficient of L_I, although quite small, is positive, indicating some benefit from following intradocument links. The ANOVA analysis for this model indicates that our independent variables are indeed related to our dependent variable. The 95 percent confidence intervals of the model coefficients show that, as with the models discussed above, we cannot reject our null hypothesis with respect to the interdocument links, but we also note the probability is high that the coefficient of L_I is 0 ($p > 0.18$).

Thus, we are led to conclude that intradocument links had no across-the-board effect on Ans for this particular questioning-answering task. This conclusion seems to be borne out by the subjects' answers on the post-task questionnaire. The average score on the question "Were the links *within* the articles useful?" was 2.9, between "Not really" and "Somewhat." Separate regression models for the High and Low Web groups, including the number of intradocument links and using Ans as the dependent variable were not significant and, in any case, the probability that the coefficient of L_I is 0 in these models is still very high.

When we consider Ans_V as our dependent variable, the model for the High Web group is still not significant, and

BNSDOCID: <XP_____2317545A_I_>

TABLE 15
Ninth-Five Percent Confidence Intervals for Coefficients in a Model Using All Three Link Types and Viewed Answers

Param	Value	Error	t	p	Low	High
L_{HT}	0.58	0.13	4.37	0.00	0.31	0.85
L_{MG}	0.21	0.13	1.62	0.06	−0.05	0.47
L_I	0.21	0.10	2.19	0.02	0.01	0.40

there is still a high probability that the coefficient of L_I is 0. For our Low Web group, who followed significantly more intradocument links than the High Web group, the model that results is significant and has the following equation:

$$Ans_V = 0.58 \cdot L_{HT} + 0.21 \cdot L_{MG} + 0.21 \cdot L_I \ (R^2 = 0.41).$$

Table 15 shows the 95 percent confidence intervals for this model. We see that the coefficient of L_I is always positive, indicating some effect on Ans_V from intradocument links. We also see that the probability that this coefficient is 0 is less than 0.02. We note, however, that for this model we cannot claim that the coefficient of L_{HT} is always greater than the coefficient of L_{MG}. This is not too surprising in light of the fact that the High Web group chose significantly more HT links than did the Low Web group.

6.3 Discussion

The most important conclusion that we can draw from the study is that the interdocument hypertext links generated by the method described in this thesis were not significantly better than links generated by a competing methodology for a questioning-answering task such as the one we posed to our subjects.

Having said this, however, we note that the probability of results such as those we achieved occurring by chance are less than 0.1. In addition, we can demonstrate at least one partition of our subjects (the Low and High Web groups) such that the only significant differences between them were the number of HT links followed and the number of answers found. This would seem to indicate some benefit from following HT links over MG links. For these reasons, we therefore conclude that it is necessary to replicate this evaluation in order to gain more evidence about the relationships between the number and kinds of interdocument links followed and the number of correct answers found.

Another interesting conclusion we draw is that, in general, the intradocument links did not have any benefit for the questioning-answering task that we designed. Only the Low Web group showed a significant benefit from using intradocument links, and then only when considering the number of viewed answers. This result is probably an indication of the novice's need for tools that make using unfamiliar information systems easier.

We believe that there were several factors that affected the study, some of which might have reduced the effectiveness of our methods, leading to our inconclusive results.

6.3.1 Implementation Factors

There are several problems with the implementation of the current system that, when fixed, would allow our method to perform even more effectively.

6.3.1.1 The Evaluation System. Foremost among these factors was the speed of the system. Even though we could generate links from a document in less than two seconds, many of the subjects felt that the system was "too slow." The speed of the system tended to limit the number of documents that a user could actually read in the 15 minutes allotted for each question. This factor was mitigated by the fact that once a document had been visited, the hypertext links leading from it were stored so that subsequent visits would be almost instantaneous.

Several subjects noted after they had finished their tasks that they did not feel that they could judge where an intradocument link would take them. Clearly, some more study is needed as to what would constitute good intradocument link anchors. Using the first few words of the target paragraph as the anchor text is a compromise position, one that is vulnerable to several effects, most notably pronouns with no referents. One possibility is to allow the user a way to "peek" at more of the target paragraph. This would be relatively easy to implement.

6.3.1.2 The Lexical Chainer. The current implementation of the lexical chainer, upon which all of our work is based, has some deficiencies. Of these, probably the most damaging is that words that do not appear in WordNet can never be included in a chain. This excludes a large class of words that are important in the newspaper domain, namely proper nouns. These words can never be used in a lexical-chain-based comparison of document similarity, even if they appear in both documents.

Perhaps a more subtle problem is that we rely on the lexical disambiguation performed by the chainer to solve the problem of polysemy. There are two ways in which a failure in this mechanism will negatively affect our document-linking capabilities. First, the chainer can incorrectly disambiguate a word, choosing a single, incorrect synset to represent it. This incorrect synset is then used in building the weighted synset vectors used for document comparison. When the vector for the document containing the incorrect synset is compared to other document vectors, some portion of the similarity of the documents will be missed. Unfortunately, there is no way to tell whether the chainer has incorrectly disambiguated a word, and we have no data on the average number of incorrect disambiguations per document.

The second kind of failure of the disambiguation mechanism is when it does not work at all (or works very badly), leaving a word that is represented by several synsets, each of which is counted when building the weighted synset vectors. This can result in spurious document connections. For example, during the evaluation, the "starter" document for question 1 contained the word *piece*, a word that is in 11 WordNet synsets. This word was not disambiguated at all. Another, totally unrelated document, suffered the same fate. On the basis of the weights of

BNSDOCID: <XP_____2317545A__I_>

these 11 synsets, the member-member similarity of these documents was 0.477. This led to these documents being linked with a highly ranked connection!

Clearly, we would like to avoid this sort of spurious connection. It is less obvious how we could avoid such things happening, but it is interesting to note that, in this particular case at least, the member-linked similarities for the two documents were both 0. A threshold on the two member-linked similarities, in addition to the threshold of 0.15 on the member-member similarities, may be enough to solve this problem. In the longer term, we believe that a more cautious approach to lexical chaining may be needed, that is, an approach that may take more time, but is less likely to make these sorts of errors.

6.3.2 Task Factors

Questioning-and-answering is a very "fuzzy" task to choose for an evaluation such as we have performed. In the IR community, the process of evaluation is generally carried out in a totally automated fashion, using collections of documents and queries with known sets of relevant documents. Of course, we could perform similar evaluations, but we are more interested in seeing how the hypertexts that we build can be used by people to perform a specific task.

Designing the questions for a task to be performed by people is not an exact science, so we have to assume that the subjects had, at best, an imperfect understanding of the questions that they were supposed to answer—even though the average response on the questionnaire to the question "I understood the questions I was supposed to answer" lay between "Agree" and "Strongly agree." This variation in understanding would obviously cause a variation in the answers that the subjects recorded. The way to avoid this seems to be to pose questions that require as little interpretation as possible on the part of the subject.

The subjects performed best on question 2, where the idea was simply to find the names of terrorists. This is a relatively straightforward task and requires little interpretation since most of the names in the database are actually identified as terrorists in the documents. In the case of the other two questions, however, some subjects seemed to have some real difficulty. For example, in more than one case, subjects answering question 3 reported only the name of the biotechnology company involved in a merger, rather than the names of all companies involved. In other cases, some subjects seemed to have difficulty distinguishing the name of a drug manufacturer from the name of the drug that they manufacture. This underscores the need for pilot testing in such evaluations.

6.3.3 The Influence of the Newspaper Domain

Newspaper articles are written so that one can stop reading them at the end of any particular paragraph and still feel as though one has read a complete story. This property of news articles may account for the performance of our intradocument links in this evaluation. If news articles are written to be skimmed, then it is likely that people will skim them. Since people will be more familiar with a newspaper than with a hypertext system and since the subjects were aware that they were reading newspaper articles, they likely read them as they would read articles in the paper. This might not have been a winning strategy for the task that we asked the subjects to perform because, if it had been, then we would probably not have found a significant difference between the number of correct answers and the number of viewed answers (although the time restrictions would account for part of this). We did, however, find that the Low Web group had some benefit from the intradocument links. This indicates that we should not just abandon the idea of intradocument links: Rather, we should investigate how these links could be used in longer texts that are not intended to be skimmed.

7 Conclusions and Future Work

7.1 The Evaluation

Our evaluation showed that we cannot reject the null hypothesis that there are no differences in the two linking methodologies. Even so, the probability of a chance result such as those that we achieved is less than 0.1. In addition, we showed that, for a particular partition of the subjects, the only significant differences were the number of HT links followed and the number of answers found. We believe that there are several implementation factors that, when remedied, will produce a significant result for our system.

We were somewhat surprised by the lackluster showing of the intradocument links in our evaluation. The best that we can say about them is that, in general, they probably had no effect on how well the subjects did in their questioning-answering tasks. It may be the case that the anchors for the intradocument links simply did not provide enough information about where a link was leading.

The fact remains, however, that the Low Web group in our evaluation followed significantly more intradocument links than the High Web group and the model shown in Section 6.2.4 demonstrates that these links probably had some benefit for these subjects. Thus, such links should be provided so that the novice users can make use of them, but an experienced user should be able to turn them off or modify how they are generated.

7.2 Lexical Chaining

There were several problems with the implementation of the lexical chainer that may have lead to less-than-optimal performance during our evaluation. These problems will be fixed in the next version of the lexical chaining software.

The most important addition that we could make to the lexical chainer is proper-noun recognition. Even a simple version of this, such as collecting words that begin with upper-case characters, would improve the capabilities of the chainer. More importantly, we can add proper names to WordNet as a sort of pseudosynset. These pseudosynsets would consist of all of the variations that we can find on a person or entity's name. For example, the proper noun *Steve Martin* and the form of address *Mr. Martin* could be referring to the same individual, and should therefore be together in a synset. This would also work for company names and their abbreviations, such as *International Business Machines* and *IBM*. Although we would expect there to be many "Mr. Martins," the disambiguation properties of the lexical chainer should help to select the right one. After each

BNSDOCID: <XP_____2317645A_J_>

set of documents has been processed, the new pseudosynsets could be written to a file to be used in successive runs. Of course, these synsets will not be linked into the WordNet hierarchy, but they will allow us to build synset-based representations using words not in WordNet.

7.3 Link Typing

One of the advantages of Allan's work [7] is that the links between portions of two texts can be given a type that reflects what sort of link is about to be followed (e.g., REVISION or CONTRAST). Although Allan could not show that users would have assigned these link types themselves, this is still very interesting work. We currently have no method for producing such typed links, but it may be the case that the relations between synsets could be used to build these links once we have used our synset weight vectors to determine whether two documents are related.

7.4 Further Evaluation

Our most recent work has been aimed toward performing evaluations of our work that are based more on traditional IR measures of performance. In particular, we've been considering document categorization tasks using both the Reuters-21578 corpus and a local corpus of hand-classified texts. Although we are just beginning these evaluations, our initial results have been favorable.

References

[1] J. Westland, "Economic Constraints in Hypertext," *J. Am. Soc. Information Science*, vol. 42, no. 3, pp. 178–184, 1991.

[2] S. Green, "Automatically Generating Hypertext by Computing Semantic Similarity," PhD thesis, Univ. of Toronto, 1997.

[3] D. Ellis, J. Furner-Hines, and P. Willett, "On the Creation of Hypertext Links in Full-Text Documents: Measurement of Inter-Linker Consistency," *J. Documentation*, vol. 50, no. 2, pp. 67–98 1994.

[4] R. Rada and D. Diaper, "Converting Text to Hypertext and Vice Versa," *Hypermedia/Hypertext and Object-Oriented Databases*, H. Brown, ed., ch. 9, pp. 167–200, Chapman and Hall, 1991.

[5] J. Morris and G. Hirst, "Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text," *Computational Linguistics*, vol. 17, no. 1, pp. 21–48, 1991.

[6] P. Thistlewaite, "Automatic Construction and Management of Large Open Webs," *Information Processing and Management*, vol. 33, no. 2, pp. 161–173, 1997.

[7] J. Allan, "Building Hypertext Using Information Retrieval," *Information Processing and Management*, vol. 33, no. 2, pp. 145–159, 1997.

[8] E. Voorhees, "Query Expansion Using Lexical-Semantic Relations," *Proc. SIGIR '94*, Dublin, ACM, July 1994.

[9] M. Halliday and R. Hasan, *Cohesion in English*. Longman, 1976.

[10] *Roget's Int'l Thesaurus*, fifth ed., R. Chapman, ed., Harper Collins, 1992.

[11] D. St-Onge, "Detecting and Correcting Malapropisms with Lexical Chains," MS thesis, Univ. of Toronto, published as Technical Report CSRI-319, 1995.

[12] R. Beckwith, C. Fellbaum, D. Gross, and G. Miller, "WordNet: A Lexical Database Organized on Psycholinguistic Principles," *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, U. Zernik, ed., pp. 211–231, Lawrence Erlbaum, 1991.

[13] D. Tudhope and C. Taylor, "Navigation via Similarity: Automatic Linking Based on Semantic Closeness," *Information Processing and Management*, vol. 33, no. 2, pp. 233–242, 1997.

[14] S. Shellenbarger, "High-Tech Parenting Virtually a Finger Tip Away," *The Globe and Mail*, p. A10, Dec. 1995.

[15] D. Ellis, J. Furner-Hines, and P. Willett, "The Creation of Hypertext Linkages in Full-Text Documents: Parts I and II," Technical Report RDD/G/142, British Library Research and Development Dept., Apr. 1994.

[16] J. Gadd, "Children's Aid Societies Plan Staff, Services Cuts," *The Globe and Mail*, p. A10, Sept. 1995.

[17] J. Gadd, "Child Aid 'On Double-Edged Sword'," *The Globe and Mail*, p. A14, Dec. 1995.

[18] G. Salton and J. Allan, "Selective Text Utilization and Text Traversal," *Proc. Hypertext '93*, pp. 131–144, ACM, Nov. 1993.

[19] A. Forsyth, "A Dictionary/Thesaurus for a Document Retrieval System," MS thesis, Univ. of Toronto, 1986.

[20] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman, "Indexing by Latent Semantic Analysis," *J. Soc. Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[21] G. Marchionini, S. Dwiggins, A. Katz, and X. Lin, "Information Seeking in Full-Text End-User-Oriented Search Systems: The Roles of Domain and Search Expertise," *Library and Information Science Research*, vol. 15, no. 1, pp. 35–69, 1993.

[22] I. Witten, A. Moffat, and T. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, 1994.

[23] B. Nordhausen, M. Chignell, and J. Waterworth, "The Missing Link? Comparison of Manual and Automated Linking in Hypertext Eng.," *Proc. Human Factors Soc. 35th Ann. Meeting*, 1991.

[24] G. Golovchinsky, "What the Query Told the Link: The Integration of Hypertext and Information Retrieval," *Proc. Hypertext '97*, pp. 67–74, ACM, Apr. 1997.

Stephen J. Green received his bachelor of mathematics degree (with honors) from the University of Waterloo in Canada in 1990. His master's work was also performed at the University of Waterloo, under the supervision of Professor Chrysanne DiMarco. His thesis, which was completed in 1992, was in the area of natural language generation. He subsequently began his PhD studies at the University of Toronto, under the supervision of Professor Graeme Hirst. His thesis work involved the automatic generation of hypertext links in large document collections, particularly large collections of newspaper articles. He completed his PhD in September 1997 and is currently a research fellow at the Microsoft Research Institute of the Division of Information and Communication Sciences at Macquarie University in Sydney, Australia. His research interests include not only work in automatic hypertext generation, but also in natural language generation.

BEST AVAILABLE COPY

BNSDOCID: <XP_____2317545A__I_>